Monitor Client NATMOPI

The Monitor Client NATMOPI is a character-based command interface for monitoring the various types of servers that can be implemented in a mainframe Natural environment.

Currently, the Monitor Client NATMOPI is available only for use in conjunction with the Natural Development Server which is part of Natural's Single Point of Development.

The following topics are covered:

- Prerequisites for NATMOPI Execution
- Command Interface Syntax
- Command Options Available
- Monitor Commands
- Directory Commands
- Command Examples
- Execute NATMOPI under TSO
- Execute NATMOPI in Batch
- Execute NATMOPI under the UNIX Shell

Prerequisites for NATMOPI Execution

NATMOPI can be executed in batch, under TSO and under the OS/390 UNIX shell. However, the user who executes NATMOPI must be defined in the OS/390 UNIX System Services.

Command Interface Syntax

Basically the syntax of the command interface consists of a list of options where each option can/must have a value. For example:

```
-s server-id -c help
```

where *-s* and *-c* are options and *server-id* and *help* are the option values.

It is possible to specify multiple options, but each option can have only one value assigned.

The command options available are listed below.

Command Options Available

Words enclosed in <> are user supplied values and words enclosed in [ ] are optional.

Command Option	Action
-s <*server-id*>	Specify a server-id for sending a monitor command. If the server-id is not unique in the server directory NATMOPI prompts the user to select a server.
-c <*monitor command*>	Specify a monitor command to be sent to the server id defined with the -*s* option
-d <*directory command*>	Specify a directory command to be executed.
-a	Suppress prompting for ambiguous server-id. Process all servers which apply to the specified server-id.
-h	Print NATMOPI help.

Monitor Commands

These are commands that are sent to a server for execution. The monitor commands available depend on the type of server, however, each server must be able to support at least the commands **terminate** and **help**. For further commands, refer to the corresponding server documentation.

Directory Commands

Directory commands are not executed by a server but directly by the monitor client NATMOPI.

You can use the directory commands to browse through the existing server entries and to remove stuck entries.

The following directory commands are available. Words enclosed in <> are user supplied values and words enclosed in [ ] are optional.

Directory Command	Action
ls [<*server-id*>]	List all servers from the server directory that are apply to the specified server-id. The server list is in short form.
ll [<*server-id*>]	Same as *ls* but the server list contains extended server information.
rs [<*server-id*>]	Remove server entries from server directory. **Note:** If you remove the entry of an active server you will loose the ability to monitor this server process.
cl [<*server-id*>]	Clean up server directory. This command pings the specified server. If the server does not respond its entry is removed from the directory.
ds	Dump the content of the server directory.
lm	List pending IPC messages.

Command Examples

`natmopi -dls`	List all servers registered at the directory in short format.
`natmopi -dcl TST -ls TST`	Clean up all servers with id TST* (ping server and remove if not responding), and list all servers with id TST* after cleanup.
`natmopi -sSRV1 -cping -sSRV2 -sSRV3 -cterminate`	Send command **ping** to SRV1. Send command terminate to SRV2 and SRV3.
`natmopi -cterminate -sSRV1 -cping -sSRV2 -sSRV3`	Is equivalent to the example above. That is NATMOPI send the command following the -s option to the server. If no -c option follows the -s the first -c option from the command line is used.
`natmopi -sSRV1 -cterminate -a`	Send command **terminate** to SRV1. If SRV1 is ambiguous in the server directory send the command to all SRV1 servers without prompting for selection.

Execute NATMOPI under TSO

Sample TSO CLIST to execute NATMOPI

```
PROC 0
CONTROL NOFLUSH ASIS LIST CONLIST
ALLOC FILE(STDOUT) DA(*)
ALLOC FILE(STDERR) DA(*)
ALLOC FILE(STDIN) DA(*)
ISPEXEC LIBDEF ISPLLIB DATASET ID('the natural load dataset' )
CALL 'the natural load dataset'(natmopi)' +
     '-sSRV1 -sSRV2 -cping +
  ' ASIS
```

Note:
The ASIS keyword at the end of the CALL parameters is required to instruct TSO to pass the command line in mixed case. The 'natural load dataset' must contain the modules NATMOPI and NATMONI.

Execute NATMOPI in Batch

The following is a sample JCL to execute NATMOPI in batch.

```
//NATMOPI JOB CLASS=K,MSGCLASS=X
//*
//RUN   EXEC PGM=NATMOPI,PARM='-SSERVER1 -SSERVER2 -Cping
//              -sNvdSrv1 -clist sessions'//*
//STEPLIB  DD  DISP=SHR,DSN=the natural load dataset
//SYSPRINT DD  SYSOUT=*
/*
```

Note:
If you are prompted in batch to select an ambiguous server, NATMOPI writes the prompt to the operator console. You have to reply to the prompt using the following operator command:


```
F jobname APPL=index
```

Execute NATMOPI under the UNIX Shell

Since the UNIX shell does not support the execution of modules which reside in a PDS, it is neccessary to link NATMOPI to the UNIX HFS.

The following is a sample JCL to link NATMOPI to the HFS.

```
//NATLNK   JOB CLASS=K,MSGCLASS=X
//LKXMON   EXEC PGM=IEWL,
//         PARM='RENT,XREF,LIST,LET,REUS,SIZE=(300K,64K),CASE=MIXED'
//SYSUT1   DD   UNIT=(SYSDA),SPACE=(TRK,(10,4))
//SYSLMOD  DD  PATH='/tmp',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=(SIRWXG,SIRWXU)
//SYSPRINT DD   SYSOUT=X
//NATLOAD  DD  DISP=SHR,DSN= the natural load dataset
//SYSLIN     DD *
 INCLUDE NATLOAD(NATMOPI)
 NAME natmopi(R)
/*
```

The following is a sample shell command to execute NATMOPI.

```
> export STEPLIB=the natural load dataset:$STEPLIB
> /tmp/natmopi -dls -sSRV1 -cping
```

Note:
The environment variable STEPLIB must contain the Natural load dataset to enable NATMOPI to load the module NATMONI dynamically.

